More Observable Systems with the elastic stack

Philipp Krenn @xeraa

elastic

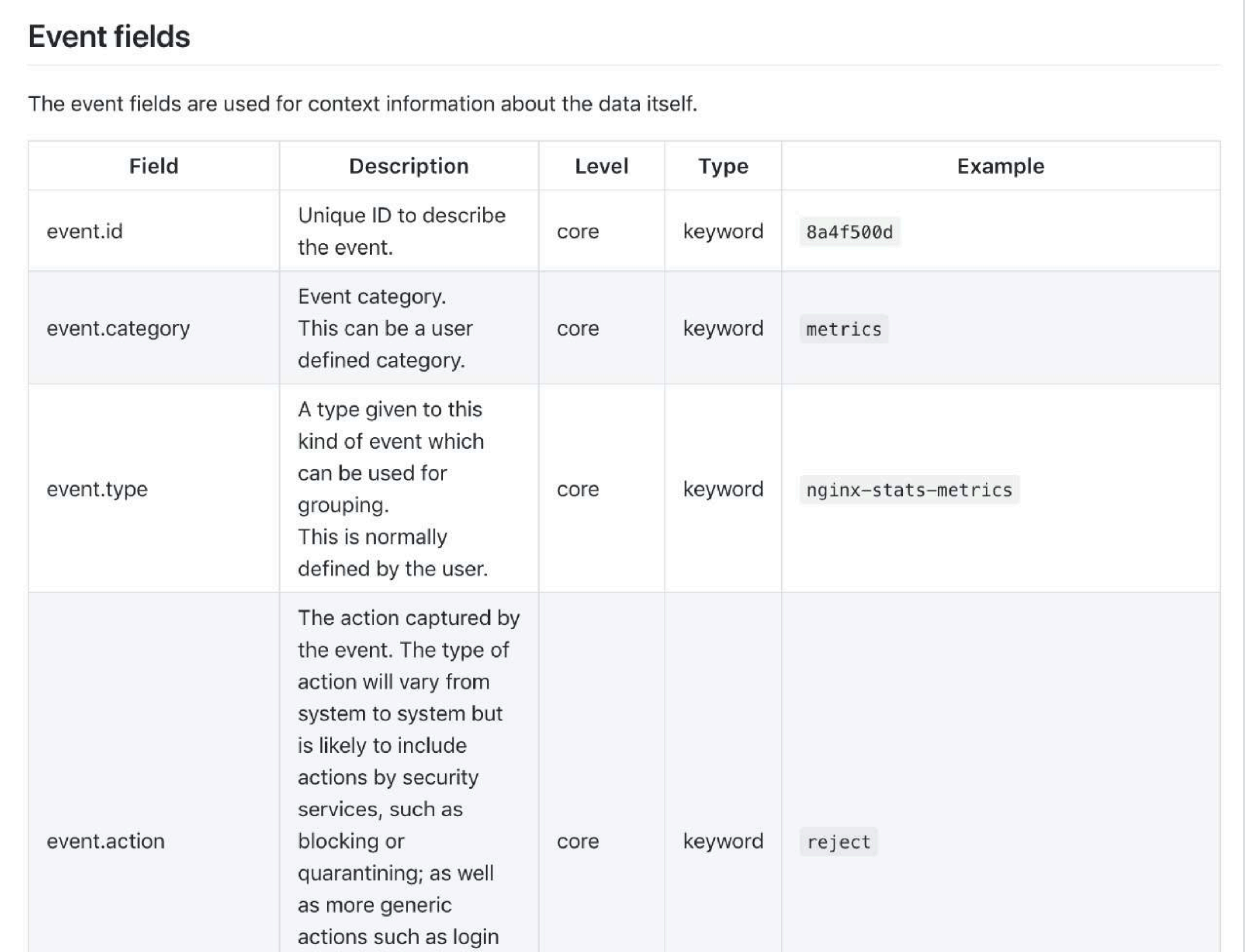
Developer 🥑

A system is observable if the behaviour of the entire system can be determined by only looking at its inputs and outputs.

— Kálmán (1961), On the General Theory of Control Systems

Logs & Elastic Common Schema (ECS)

https://github.com/elastic/ecs



Event fields

The event fields are used for context information about the data itself.

Field	Description	Level	Type	Example
event.id	Unique ID to describe the event.	core	keyword	`8a4f500d`
event.category	Event category. This can be a user defined category.	core	keyword	`metrics`
event.type	A type given to this kind of event which can be used for grouping. This is normally defined by the user.	core	keyword	`nginx-stats-metrics`
event.action	The action captured by the event. The type of action will vary from system to system but is likely to include actions by security services, such as blocking or quarantining; as well as more generic actions such as login	core	keyword	`reject`

The classic way of logging

Logstash & grok

The new way of logging

https://github.com/elastic/ecs-logging-java/

Plus PHP & .NET — more coming

From log...

```
{
  "@timestamp":"2019-11-28T19:36:16.872Z",
  "log.level": "WARN",
  "message":"[philipp] failed to log in with password [***]",
  "service.name":"gs-securing-web",
  "process.thread.name":"http-nio-8080-exec-5",
  "log.logger":"hello.AuthenticationEventListener"
}
```

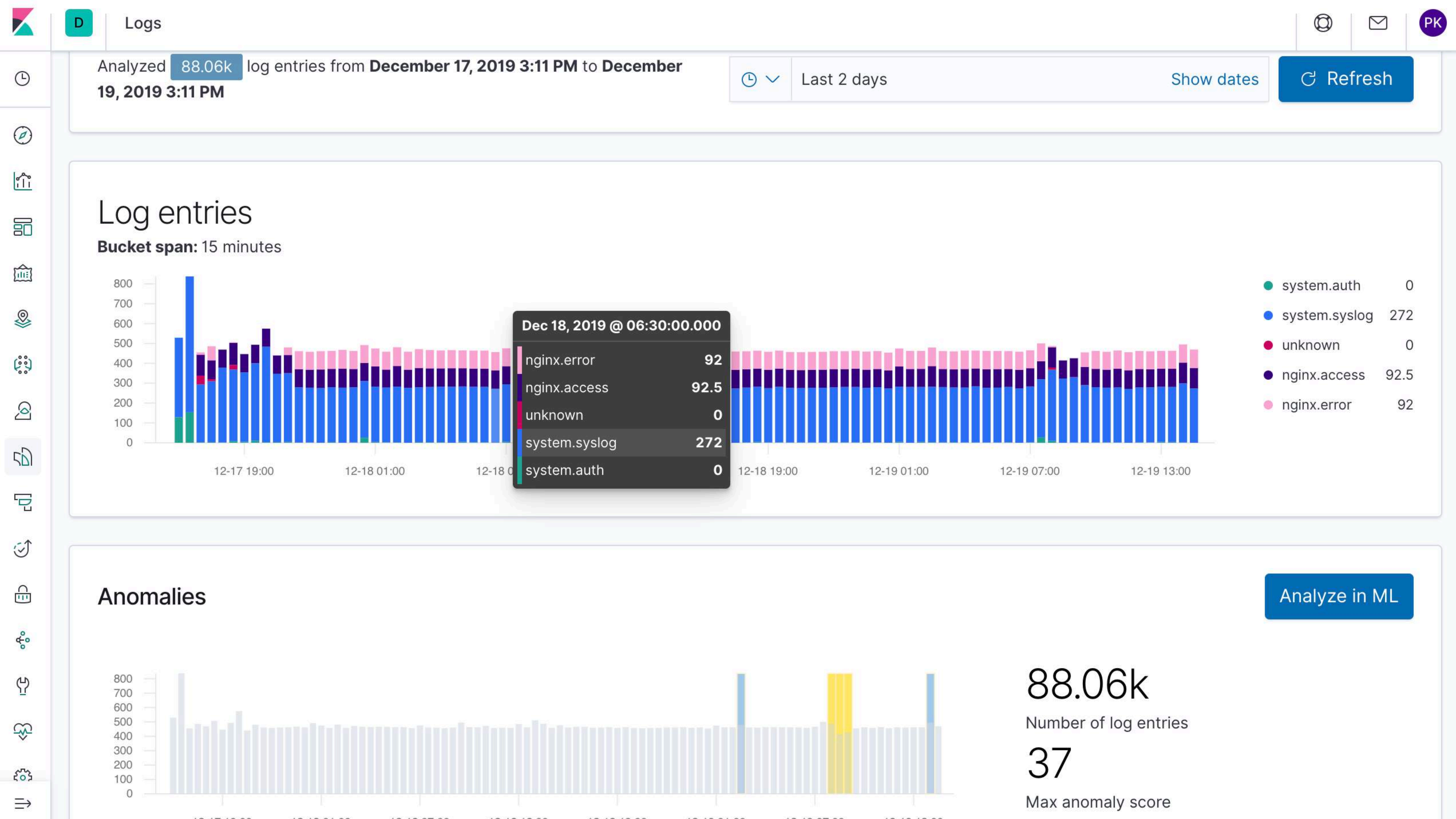
...to event

```
{
   "@timestamp":"2019-11-28T19:36:16.872Z",
  "log.level": "WARN",
  "message":"[philipp] failed to log in with password [***]",
  "service.name":"gs-securing-web",
  "process.thread.name":"http-nio-8080-exec-5",
  "log.logger":"hello.AuthenticationEventListener",
  "labels.event.category": "LOGIN_FAILURE",
  "labels.user.name": "philipp",
  "labels.source.ip": "0:0:0:0:0:0:0:1",
  "labels.url.full": "/login"
}
```

PS: Easy visualization with Lens

Log rate anomaly detection

New, higher, or no logs



Analyzed 88.06k log entries from **December 17, 2019 3:11 PM** to **December 19, 2019 3:11 PM**

Log entries

Bucket span: 15 minutes

Dec 18, 2019 @ 06:30:00.000

nginx.error	92
nginx.access	92.5
unknown	0
system.syslog	272
system.auth	0

● system.auth	0
● system.syslog	272
● unknown	0
● nginx.access	92.5
● nginx.error	92

Anomalies

88.06k

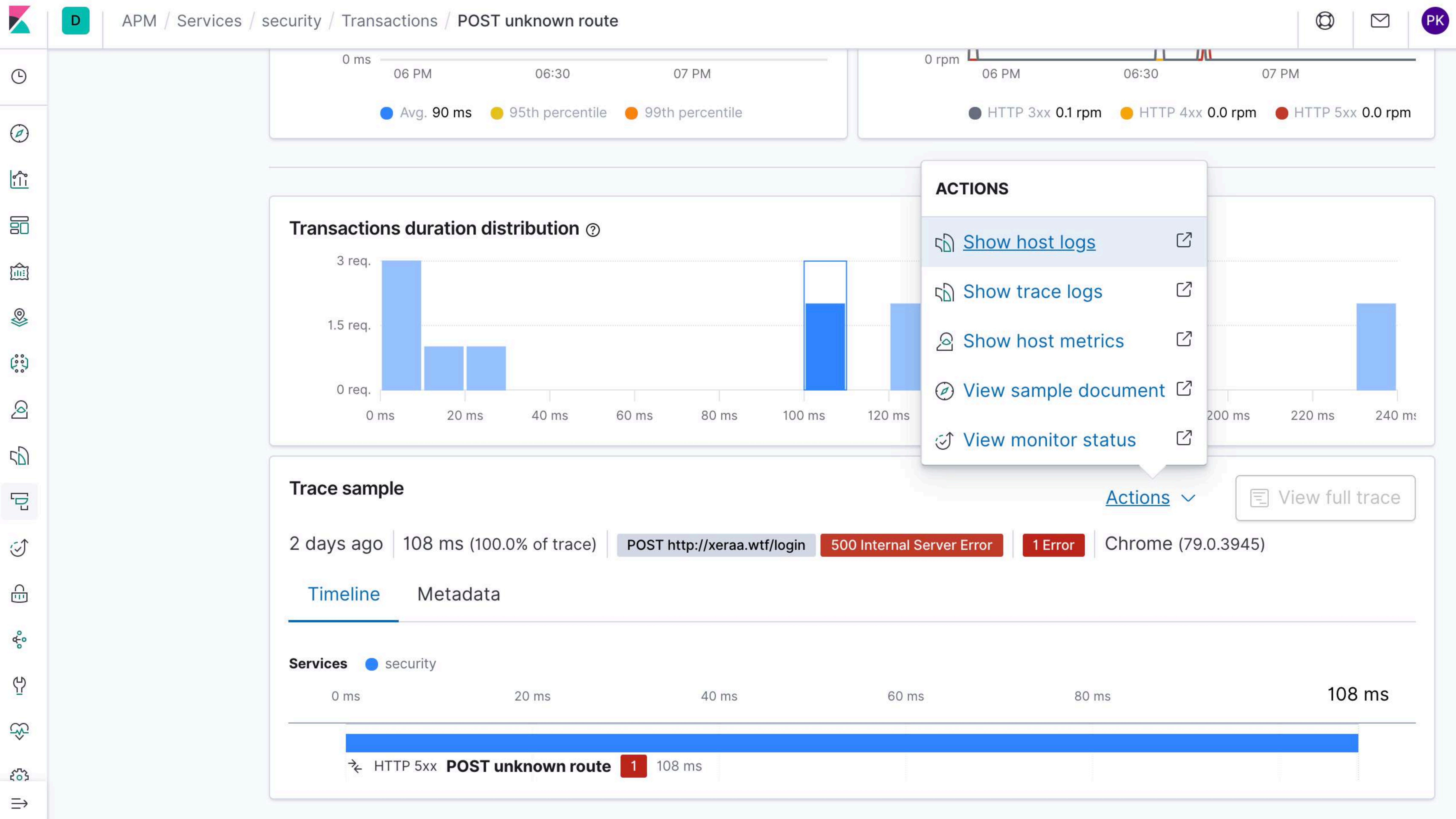
Number of log entries

37

Max anomaly score

Logs ❤️ Traces

trace-ids in common logging frameworks from Elastic APM agents



Transactions duration distribution

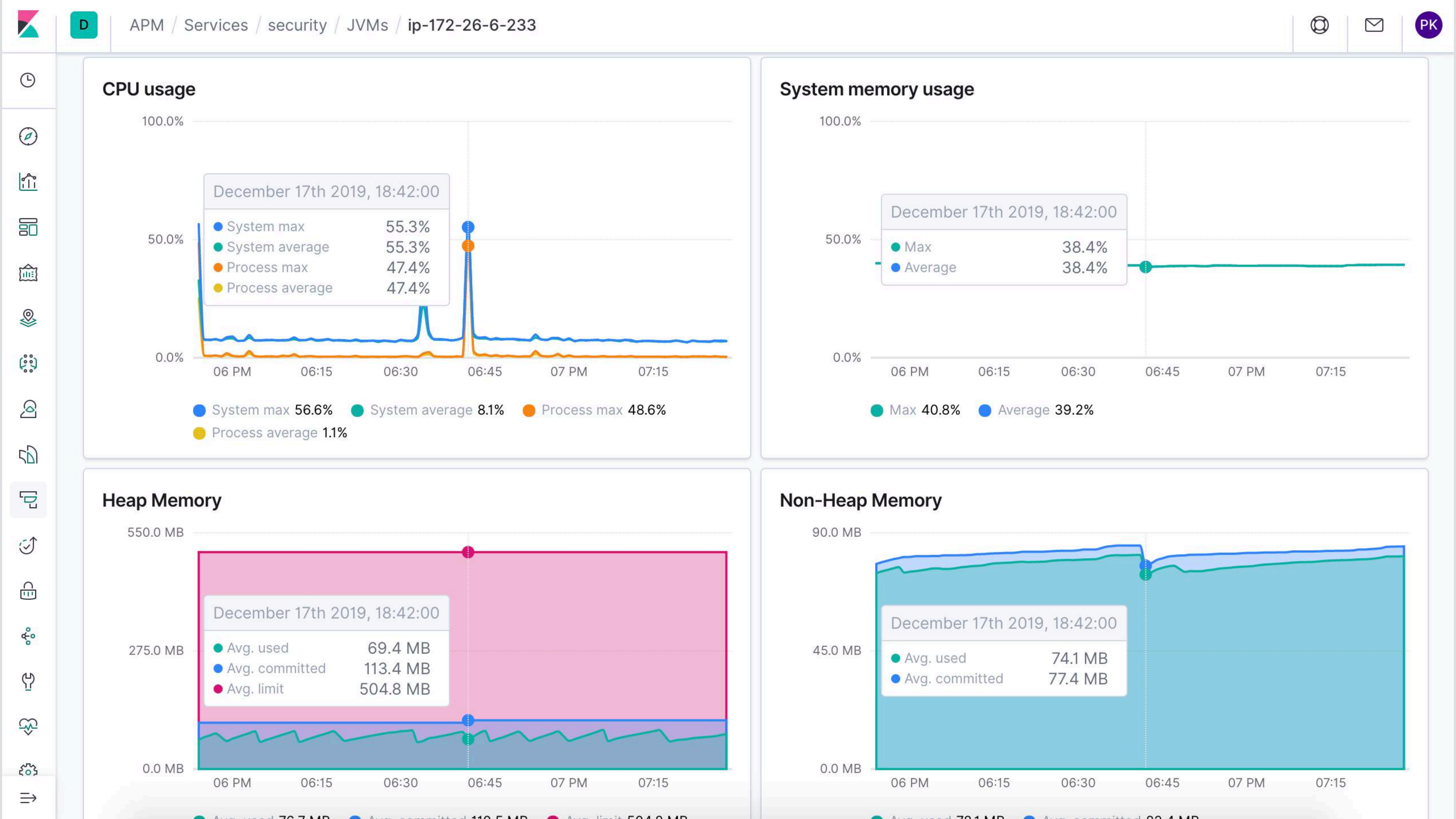
Trace sample

2 days ago | 108 ms (100.0% of trace) | POST http://xeraa.wtf/login | 500 Internal Server Error | 1 Error | Chrome (79.0.3945)

HTTP 5xx **POST unknown route** 1 108 ms

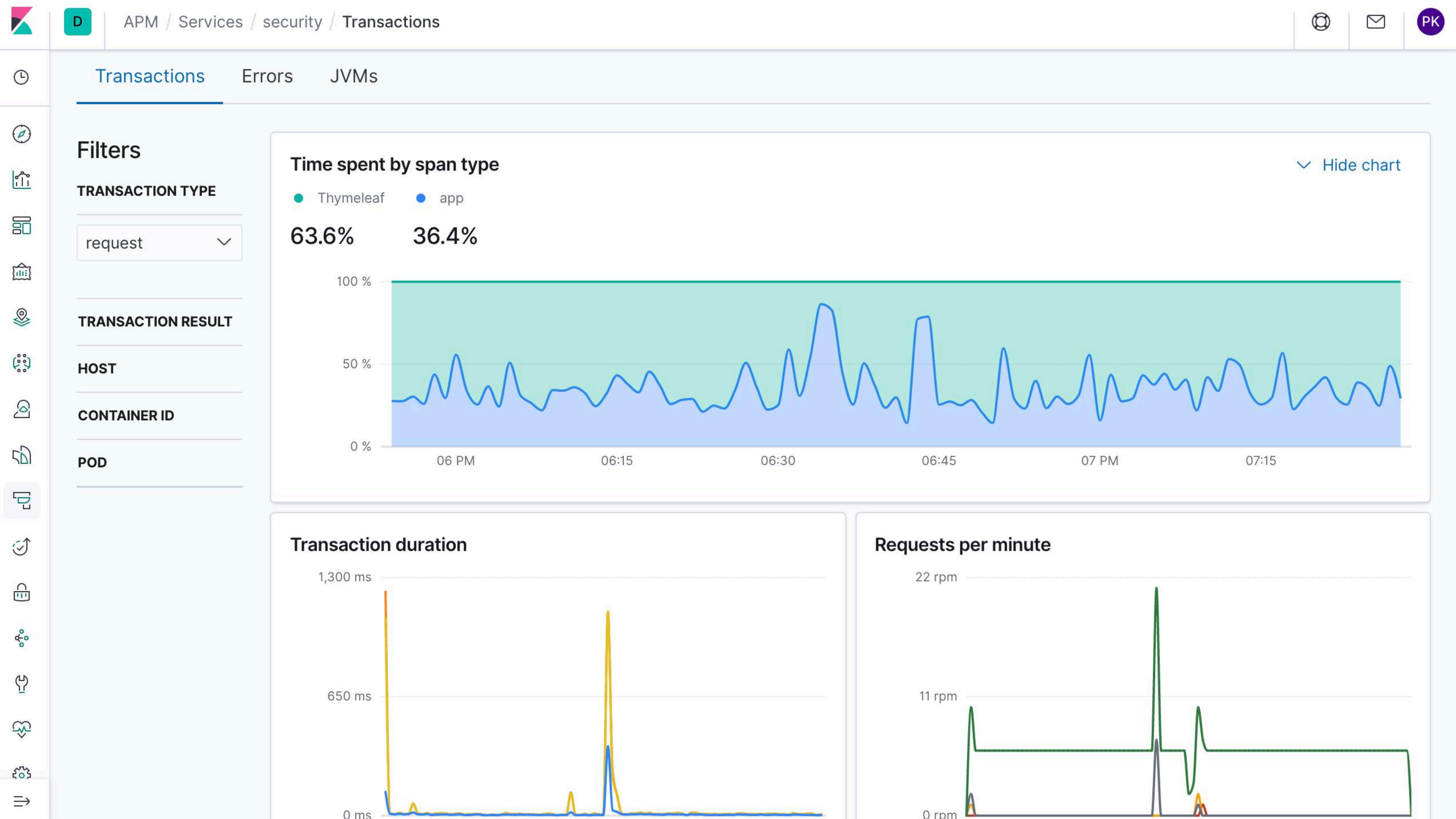
JVM metrics

Instance-level visibility and metadata


D
APM / Services / security / JVMs / ip-172-26-6-233
PK
CPU usage
100.0%
50.0%
0.0%
December 17th 2019, 18:42:00
System max 55.3%
System average 55.3%
Process max 47.4%
Process average 47.4%
06 PM
06:15
06:30
06:45
07 PM
07:15
System max 56.6%
System average 8.1%
Process max 48.6%
Process average 1.1%
System memory usage
100.0%
50.0%
0.0%
December 17th 2019, 18:42:00
Max 38.4%
Average 38.4%
06 PM
06:15
06:30
06:45
07 PM
07:15
Max 40.8%
Average 39.2%
Heap Memory
550.0 MB
275.0 MB
0.0 MB
December 17th 2019, 18:42:00
Avg. used 69.4 MB
Avg. committed 113.4 MB
Avg. limit 504.8 MB
06 PM
06:15
06:30
06:45
07 PM
07:15
Avg. used 76.7 MB
Avg. committed 110.5 MB
Avg. limit 504.8 MB
Non-Heap Memory
90.0 MB
45.0 MB
0.0 MB
December 17th 2019, 18:42:00
Avg. used 74.1 MB
Avg. committed 77.4 MB
06 PM
06:15
06:30
06:45
07 PM
07:15
Avg. used 78.1 MB
Avg. committed 82.4 MB

Service breakdown chart

Aggregated time spent in code, database, external calls



APM / Services / security / Transactions

Transactions Errors JVMs

Filters

TRANSACTION TYPE

request

TRANSACTION RESULT

HOST

CONTAINER ID

POD

Time spent by span type

Hide chart

● Thymeleaf 63.6%

● app 36.4%

Transaction duration

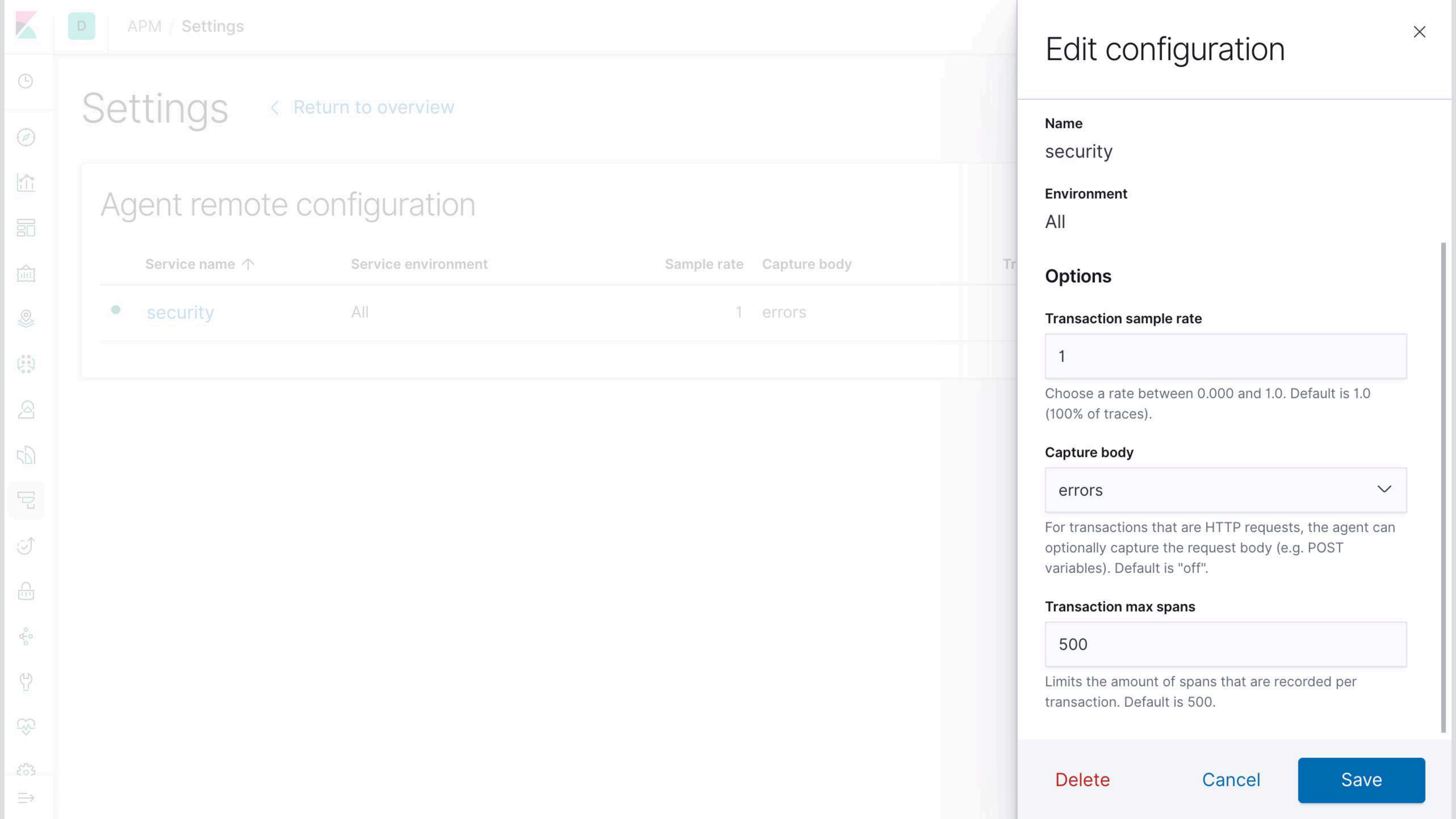
Requests per minute

APM agent central management

Capture request body

Number of spans

Deployment status



Settings

Agent remote configuration

	Service name ↑	Service environment	Sample rate	Capture body	Tr
●	security	All	1	errors	

Edit configuration

Name

security

Environment

All

Options

Transaction sample rate

1

Choose a rate between 0.000 and 1.0. Default is 1.0 (100% of traces).

Capture body

errors

For transactions that are HTTP requests, the agent can optionally capture the request body (e.g. POST variables). Default is "off".

Transaction max spans

500

Limits the amount of spans that are recorded per transaction. Default is 500.

Cloud Infrastructure

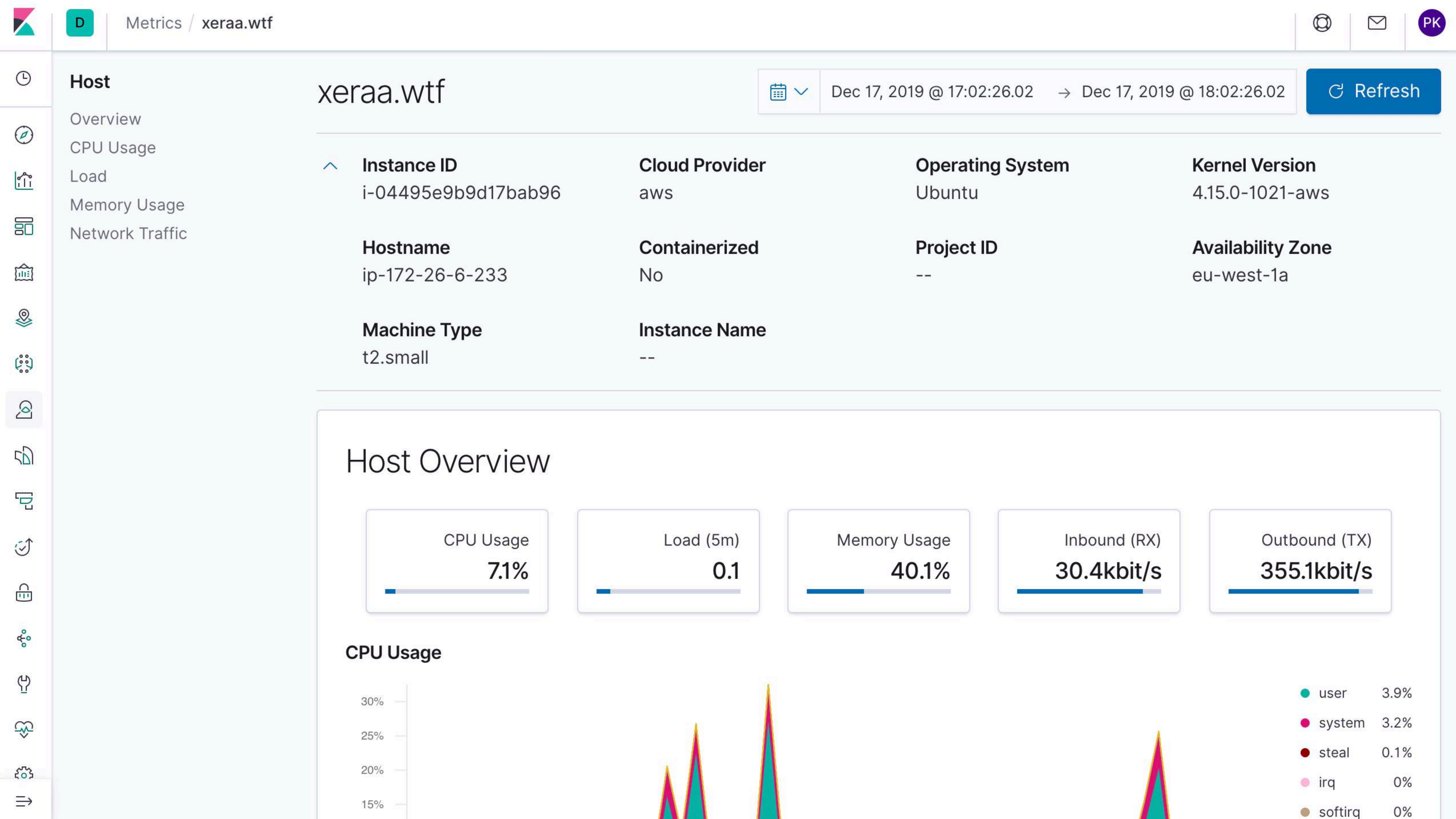
Azure (new) & AWS (improved) monitoring

Azure Metricsets

monitor

compute_vm

compute_vm_scaleset



xeraa.wtf

Dec 17, 2019 @ 17:02:26.02 → Dec 17, 2019 @ 18:02:26.02

Instance ID
i-04495e9b9d17bab96

Cloud Provider
aws

Operating System
Ubuntu

Kernel Version
4.15.0-1021-aws

Hostname
ip-172-26-6-233

Containerized
No

Project ID
--

Availability Zone
eu-west-1a

Machine Type
t2.small

Instance Name
--

Host Overview

CPU Usage	Load (5m)	Memory Usage	Inbound (RX)	Outbound (TX)
7.1%	0.1	40.1%	30.4kbit/s	355.1kbit/s

CPU Usage

● user	3.9%
● system	3.2%
● steal	0.1%
● irq	0%
● softirq	0%

AWS Elastic Load Balancer (ELB) logs

Request, backend, and response processing time

Connection Time

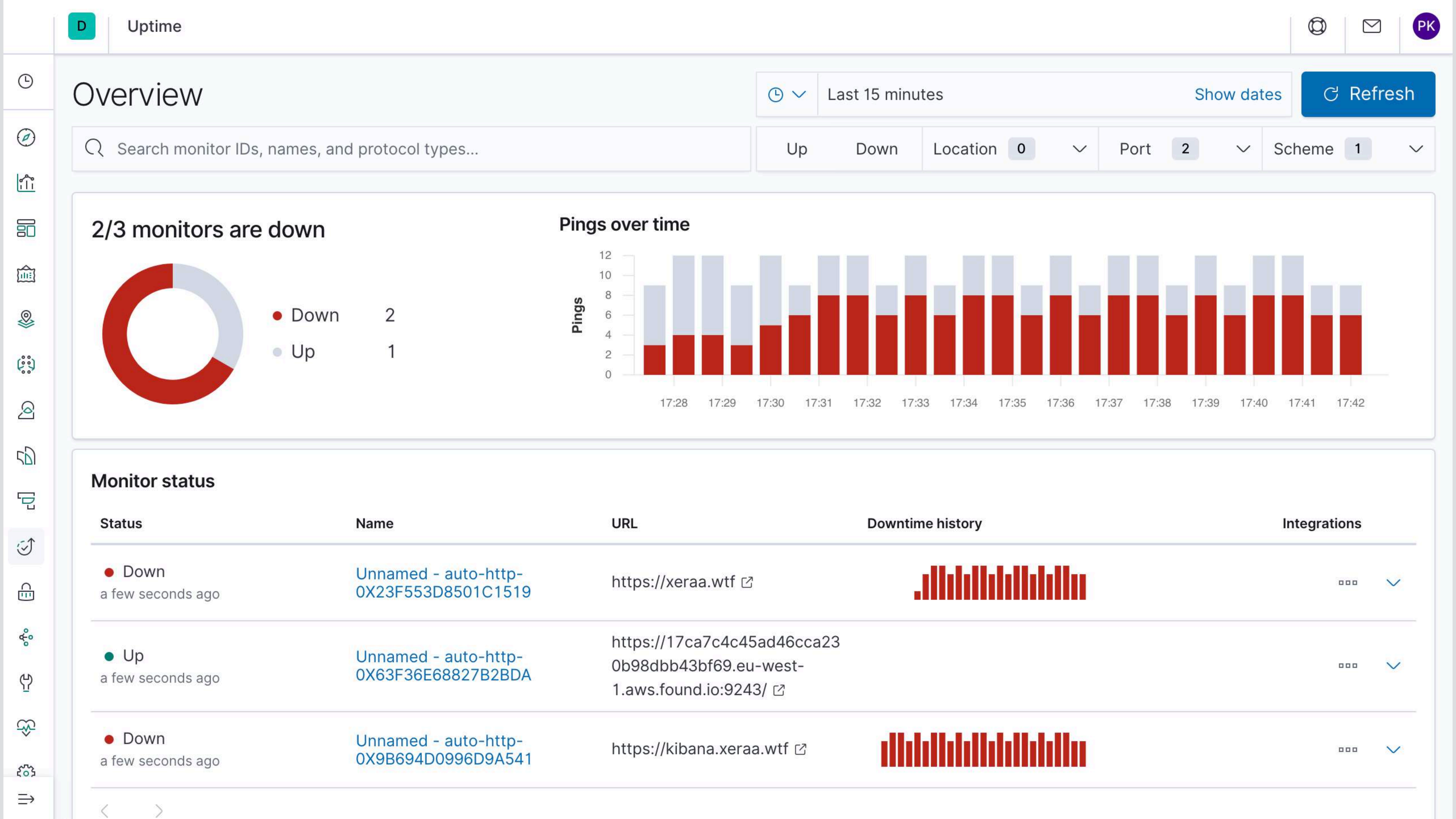
TLS Handshake Time

Health Checks

Hint-based Kubernetes autodiscovers

```
heartbeat.autodiscover:
  providers:
  - type: docker
    hints.enabled: true
```

```
LABEL co.elastic.monitor/1.type=tcp
      co.elastic.monitor/1.hosts='${data.host}:6379'
      co.elastic.monitor/1.schedule='@every 10s'
LABEL co.elastic.monitor/2.type=icmp
      co.elastic.monitor/2.hosts='${data.host}'
      co.elastic.monitor/2.schedule='@every 10s'
```



Overview

2/3 monitors are down

● Down	2
● Up	1

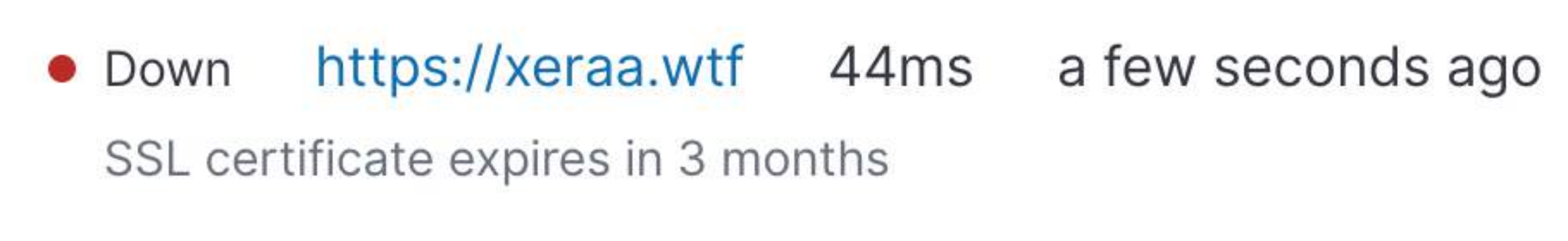
Monitor status

Status	Name	URL	Downtime history	Integrations
● Down a few seconds ago	Unnamed - auto-http-0X23F553D8501C1519	https://xeraa.wtf		
● Up a few seconds ago	Unnamed - auto-http-0X63F36E68827B2BDA	https://17ca7c4c45ad46cca230b98dbb43bf69.eu-west-1.aws.found.io:9243/		
● Down a few seconds ago	Unnamed - auto-http-0X9B694D0996D9A541	https://kibana.xeraa.wtf		

KQL & pagination support

Also better merging of multiple pingers

TLS expiration



● Down https://xeraa.wtf 44ms a few seconds ago

SSL certificate expires in 3 months

Non-privileged ICMP checks

Often supported to run without pings without root

Conclusion

You cannot buy observability...

...but tool are essentail to create observable systems

More Observable Systems with the elastic stack

Philipp Krenn @xeraa